PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06K 9/18	A1	(11) International Publication Number: WO 96/39677 (43) International Publication Date: 12 December 1996 (12.12.96)

(21) International Application Number: PCT/US96/05884

(22) International Filing Date: 26 April 1996 (26.04.96)

(30) Priority Data:
08/465,502 5 June 1995 (05.06.95) US

(71) Applicant: MOTOROLA INC. [US/US]; 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

(72) Inventor: PARTHASARATHY, Kannan; 3316 St. Michael Drive, Palo Alto, CA 94306 (US).

(74) Agents: POWELL, Jordan, C. et al.; Motorola Inc., Intellectual Property Dept., 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

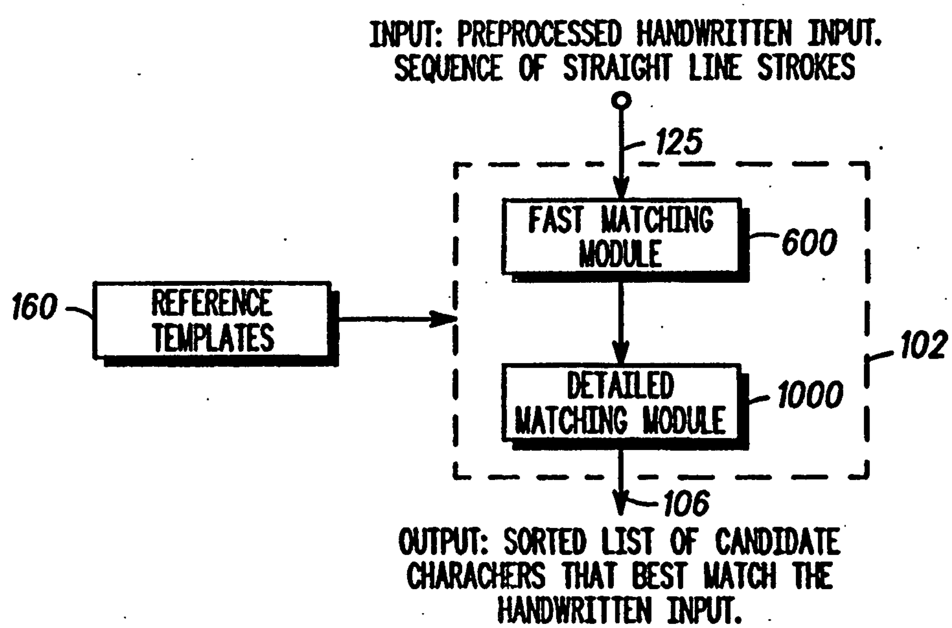
(81) Designated States: AU, CA, CN, JP, KR, SG, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: METHOD AND APPARATUS FOR CHARACTER RECOGNITION OF HAND-WRITTEN INPUT

(57) Abstract

A method and apparatus for recognition of hand-written input is disclosed where hand-written input composed f a sequence of (x, y, pen) points (125) is preprocessed into a sequence of strokes (122). A short list of candidate characters that are likely matches for the hand-written input is determined by finding a fast matching distance (600) between the input sequence of strokes and a sequence of strokes representing each candidate character of a large character set (160) where the sequence of strokes for each candidate character is derived from statistical analysis of empirical data. A final sorted list of candidate characters which are likely matches for the hand-written input (106) is determined by finding a detailed matching distance between the input sequence of strokes and the sequence of strokes for each candidate character of the short list (1000). A final selectable list f candidate characters is presented t a user.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgystan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic			SE	Sweden
CG	Congo	KR	Republic of Korea	SG	Singapore
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LR	Liberia	SZ	Swaziland
CS	Czechoslovakia	LT	Lithuania	TD	Chad
CZ	Czech Republic	LU	Luxembourg	TG	Togo
DE	Germany	LV	Latvia	TJ	Tajikistan
DK	Denmark	MC	Monaco	TT	Trinidad and Tobago
EE	Estonia	MD	Republic of Moldova	UA	Ukraine
ES	Spain	MG	Madagascar	UG	Uganda
FI	Finland	ML	Mali	US	United States of America
FR	France	MN	Mongolia	UZ	Uzbekistan
GA	Gabon	MR	Mauritania	VN	Viet Nam

METHOD AND APPRATUS FOR CHARACTER RECOGNITION OF HANDWRITTEN INPUT

Field of the Invention

This invention relates generally to handwriting recognition, and more particularly to recognition of large characters sets where each character includes one or more strokes.

Background of the Invention

Machine recognition of human hadnwriting is a very difficult problem, and with the recent explosion of pen-based computing and electronic devices, has become an important problem to be addressed. There exists various different computing and electronic devices that accept handwritten input. So called pen-based products, for example, computers, and personal digital assistants, and the like typically have a touch sensitive screen upon which a user can impose handwriting. These devices then function to digitize the handwritten input Other devices, such as computers, advanced telephones, digital televisions, and other information processing devices, can access a digitizing tablet which can accept handwritten input. Still other devices can receive handwritten character input by means of a fax, scanned input, electronic mail, or other electronic transmission of data. These devices process the information and attempt to recognize the information content of the handwritten input; Typically, the device then displays that information to the user for purposes of feedback, correction of errors in the processing, and for recognition of the handwritten character input.

There exists various approaches for recognition of handwritten input when the recoginition is for characters sets having a limited finite number of characters, typically under a

hundred. However often such approaches do not work as well for character sets having large numbers of varied complex characters. Examples of large character sets that have been difficult to quickly and accurately recognize through recognition of handwritten input are several of the asian ideograghic character/symbol languages, such as Chinese, simplified and traditional, Japanese, and other languages having large character sets. Some languages such as simplified Chinese consist of several thousand characters.

Traditional methods, such as keyboard entry, of inputing data and text supplied in one of these types of large character based languages is often very difficult; inpart because of the large number and complexity of the character set. Additionally, many of these such languages resort to phonetic based representations using Western characters in order to enter the characters with a keyboard. Hence, keyboard-type entry of such characters is difficult. An example of the difficulty of keyboard entry for a large character set based language is keyboard entry of the Chinese language. To enter data, or text, in Chinese, via a keyboard, the language is first Romanized. Western Characters, such as the English, ango-saxon alphabet are used to phonetically represent the characters of the Chinese language. This is referred to as Pin-yin. Therefore, for a person wishing to enter data or text in Chinese through a keyboard, the person must first know Pin-yin, and the corresponding English character representation for the phoentic equivalent of the chinese character they are trying to enter via the keyboard.

Another, difficulty encountered with recognition of handwritten input of data, or text, based upon a langague having a large character set is diversity among various persons is great because of the large amount of characters and the complexity of the characters themselves. Additionally, many of these such languages have one or more forms of representing the same character, similar to print and cursive forms for the

English, anglo-saxon alphabet. Additionally, such languages may have homophones for example, the Chinese language has numerous homophones - words that are pronounced the same but have different meaning and written forms. Hence, the same Pin-yin can refer to a multiplicity of characters and the person entering Chinese character data must often select from a list of possible choices.

Typically, techniques used for handwriting recognition of the english, anglo-saxon alphabet character set, or other such limited finite character sets of under hundred, do not produce accurate results for languages having large character sets, of several hundred or several thousand varied complex characters. Many of the techniques used for handwritting recognition of small character set languages are very slow when used for large character set languages.

Therefore, because of the increasing use of pen-based electronic input devices, the difficulty of keyboard entry for large, complex, character set languages, a need exsists for a method and apparatus for recognition of handwritten input for complex, large character set langauages that is quick, accurate, and easy to use.

Brief Description of the Drawings

Fig. 1 Illustrates a block diagram of operation of a preferred embodiment of the presention invention.

Fig. 1a Illustrates a top plan view of an illustrative pen-based microprocessor entry device suitable to receive input in accordance with the present invention.

Fig. 2 Illustrates a block diagram detailing operation of a preferred embodiment of the present invention.

Fig. 3 Illustrates a format of a preferred embodiment of reference templates in accordance with the present invention.

Fig. 4. Illustrates a block diagram of operation of a preferred embodiment of character matching in accordance with the present invention.

Fig. 5 Illustrates a block diagram of operation of a preferred embodiment of fast matching in accordance the present invention.

Fig. 6 Illustrates a flow diagram of operation detailing a preferred embodiment of fast matching in accordance with the present invention.

Fig. 7 Illustrates graphically a preferred embodiment of fast matching in accordance with the present invention.

Fig. 8 Illustrates graphically a preferred embodiment of fast matching in accordance with the present invention.

Fig. 9. Illustrates a block diagram for a preferred embodiment of detailed matching in accordance with the present invention.

Fig. 10. Illustrates a flow diagram of a preferred embodiment of detailed matching in accordance with the present invention.

Fig. 11. Illustrates a flow diagram of a preferred embodiment of detailed matching in accordance with the present invention.

Fig. 12. Illustrates a top plan view of an illustrative pen-based electronic entry device having a microprocessor upon which handwritten input has been received and a corresponding detailed matching has been displayed in accordance with a preferred embodiment of the present invention.

Detailed Description of Preferred Embodiment

Generally, the present invention relates to a method and apparatus for recognition of handwritten input; and preferrably the present invention relates to a method and apparatus for recognition of handwritten input representing one or more

characters selected from a language or compilation of data having a large complex set of characters where each character includes one or more strokes.

Pursuant to a preferred embodiment of the present invention, candidate characters in support of a handwritting recognition method and apparatus of the present invention are developed through the compliation and statistical analysis of empirical data compiled from hundreds of samples of actual handwritten characters. Candidate characters produced through the development of templates dervived from the compliation and statistical ananlysis of the empiracal data are selectable as the recognized character of the handwritten input.

Referring now to the Figures, Figs. 1 and 1a illustrate general operation of a method and apparatus in accordance with a preferred embodiment of the present invention. With reference to Fig. 1a, an example of a pen-based electronic entry device is illustrated. A personal digital assistant is illustrated as generally depicted by the reference numeral 10. The personal digital assistant (10) depicted constitutes a generic representation, typically such devices include a housing (12) and a touch screen (18) upon which input can be handwritten using an appropriate hand manipulated stylus (15). Such devices typically include one or more microprocessors or other digital processing devices. As such, these devices comprise computational platforms that can be readily adapted in accordance with the teachings presented herein. It should be understood that, while such personal digital assistants comprise a ready platform to accomadate the practice of the applicant's teachings, the teachings presented herein may be practiced in a variety of other operating environments as well. Some examples of such environments include, but are not limited to the following, computers or other electronic entry devices with digitizing screens, or connected to a digitizing input surface, or capable of receiving faxed, scanned, or other electronic input,

digital or interactive televisions, modems, telephones, pagers, or other systems with the ability to capture handwritten input and process it.

Referring now to Fig. 1 a block diagram of a preferred embodiment of recognizing handwritten input in accordance with the present invention is illustrated. Handwritten input in accordance with a preferred embodiment of the present invention is represented as a sequence of (x,y,pen) values where x and y represent the (x,y) coordinates of an ink point with respect to some coordinate system and pen is a binary variable that represents the pen state with respect to the input surface of a device. A pen value can either be a pen-up (pen is not in contact with the writing or input surface) or a pen-down (pen is in contact with the writing or input surface). In accordance with the present invention, handwritten input may be captured electronically using a digitizing tablet, or alternatively may be derived from a scanned or faxed image through a process of line detection in the image. Such methods of capturing handwritten input electronically are understood in the art. In a preferred method, handwritten input is accepted by a device, such as a personal digital assistant (PDA) or other device. Other devices that function to receive handwritten input include, but are not limited to, the following: computers, modems, pagers, telephones, digital televisions, interactive televisions, devices having a digitizing tablet, facsimile devices, scanning devices, and other devices with the ability to capture handwritten input.

In the present invention, the handwritten input (ink) that is presented to the recognizer corresponds to that of a single character. If two or more characters need to be recognized, then the ink corresponding to each character must be supplied to the recognizer separately in time and preferably in the desired sequential order in order to determine the identity of and preferred order of each of the characters.

In accordance with the present invention, generally the recognizer (102) performs a series of operations on the input ink (104) and produces a list of candidate characters (106) that correspond to and represent the handwritten input (20). Preferrably a list of candadate characters is provided from which selection is made of the candidate character that most likely corresponds to and represents the handwritten input. The list may be variable in the number of candidate characters presented to choose from. The candidate character that most represents and corresponds to the handwritten input can then be selected. The selection can occur through various methods, including but not limited to such methods as user selection, or language modeling, or the like. In accordance with a preferred embodiment of the present invention the recognizer of the present invention is adapted and configured to recognize individual characters which are a part, or subset, of large character sets where preferably each character includes one or more strokes and the character set comprises hundreds or even thousands of individual characters, and more preferably whose individual characters have a preponderance of straight line pieces. Examples of such large character sets include but are not limited to the the ideographic character symbols of several of the asian languages, includeing but not limited to Chinese, Japanese, etc. In accordance with a preferred method and embodiment of the present invention, recognition of [illegible] with the present invention is [illegible] with respect to the character set of the simplified Chinese language, in particular, of the characters defining the catagory of GB1 simplified Chinese characters.

Referring now to Fig. 2, a block diagram of a preferred method and apparatus is illustrated. As shown in Fig. 2, a preferred embodiment of the present invention includes access to a preprocessing module (122), a character matching module (140), and a set of reference templates (160). Preferably, the preprocessing module converts the handwritten input(20), or

raw input data, i.e. sequence of (x,y,pen) (104) values into a sequence of "strokes." In accordance with the present invention a "stroke" is defined as a basic unit of pen movement. Any handwritten input can then be represented as a sequence of strokes. A preferred repressentation for a "stroke" is a straight line parametrized is by using a four dimensional vector having the following dimensions: 1) mx, 2) my, 3) len, 4) ang. Where mx is the x coordinate of the mid point [illegible] is the y coordinate of the mid [illegible] length of the straight line stroke, and ang [illegible] the straight line stroke with [illegible] (like the x axis). Alternativ[illegible] other parametrizations of straigh[illegible] accordance with the pre[illegible] invention. The preprocessing module (122) reduces the amount of data that needs to processed by the recognizer (102) and it also serves to correlate multiple instances of the same handwritten character that look similar, and provides the recognizer (102) with a preferred quality of input. An embodiment of the prepocessing module is descibed in related U.S. patent application entitled METHOD AND MICROPROCESSOR FOR PREPROCESSING HANDWRITING HAVING CHARACTERS COMPOSED OF A PREPONDERANCE OF STRAIGHT LINE SEGEMENTS filed concurrently, and on the same day as the present application, having U.S. serial number (yet to be determined.)

In the preferred method and embodiment of the present invention, the recognizer includes the character matching module (140). Generally, The character matching module (140) correlates and compares the handwritten input of the present invention to one or more sets of stored reference templates (160) and then provides a corresponding list of prefered candidate characters that have the most probablity of representing the original handwritten input(20).

Generally, the reference templates (160) of the present invention include one or more templates or sets of templates for each character in the large character set. For a preferred embodiment of the present invention the recognizer (102) was implented for Simplified Chinese Characters (the character set used in Mainland China) and has a vocabulary size of 3755 characters. This character set is commonly referred to as GB1, where simplified Chinese Characters consist of character sets GB1 and GB2. In the preferred embodiment of the present invention several templates are referred to, or accessed, by the recognizer (102) for each character in the preferred GB1 Character set vocabulary. The multiple templates for each character are provided to balance diversity among writer to writer variations of the complex characters; and balance diversity of forms, i.e. print vs cursive, of representing the same character even among the same writer. Pursuant to a preferred embodiment of the present invention the reference templates (160) of candidate characters are developed from emphircal data The empirical data is compiled and statistical analysis is performed on hundreds of samples of actual handwritten input representing each of the potential characters to be recognized. These candidate characters are then provided in accordance with the present invention and are selectable as the corresponding and representative recognized character of the handwritten input.

[illegible] templates (160) is [illegible] that is produced by the [illegible] module (122). Accordingly, each "stroke" is parametrized in some fashion. In accordance with the present invention a "stroke" is simply a straight line parametrized is some fashion. As discussed previously a preferred way of parametrizing a "stroke" is by using a four dimensional vector having the following dimensions: 1) mx, 2) my, 3) len, 4) ang. Where mx is the x coordinate of the mid point of the stroke, my is the y coordinate of the mid point of the stroke, len is the

length of the "stroke", and ang is the angle made by the "stroke" with respect to some reference axis (like the x axis). Referring to the preferred reference templates accessed by the present invention, each "stroke" in addition to being parametrized in some fashion, which indicates how im character; the from analysis of the empi

The reference empirical data and performing for storing the reference templates in the present invention is illustrated in Fig. 3; where the number of characters in the vocabulary is denoted by M. For each character, (example, character 1 marked 302), the number of templates (304) is stored. For each template (306), the number of "strokes" (308) is then stored. For each "stroke" (310), a parametrized description of the stroke and the weight associated with the stroke is stored (312). In 312, the preferred parametrization, i.e. the four dimensional vector [mx, my, len, ang] is shown. However, other parametrizations may also be used. Alternative parameterizations of "stroke" may used in accordance with both the preprocessing module (122) of the present invention and the reference templates (160) of the present invention. However, in a most preferred embodiment of the present invention, the parameteriztion of the "stroke" are the same for the preprocessing module (122) and for the reference templates (160).

Referring now to Fig. 4, a block diagram of operation of a preferred embodiment of character matching is shown. In the preferred embodiment illustrated, the character matching module (140) includes two distinct components. The components, a fast matching module (600), and a detailed matching model (1000) of the character matching module (140) are shown in Fig. 4. Preferrably, the input to the character matching module is the sequence of straight line strokes (125)

that is produced by the selected preprocessing module (122). The "strokes" (125) represent the preprocessed handwritten input (20). The first stage, or component, of the character matching module (140) is the a fast matching module (600). Generally, the fast matching module (600) component functions to quickly provide a short list (625) of candidate characters that most likely includes a corresponding and representative match to the handwritten input (20). The second stage or component of the character matching module (140) is the detailed matching model (1000). Generally,. the detailed matching module (1000) functions to provide a detailed matching of the handwritten input (20) with only those reference templates (160) of candidate characters provided on the short list (625) produced by the fast matching module (600). Preferably, the short list (625) produced from the fast matching module (600) ,ensures that the detailed matching module can quickly and accurately provide a corresponding representative candidate character to the handwritten input. More preferably, the combintion of the fast matching module (600) and the detailed matching module (1000) provide a metod and apparatus for recognition of handwritten input that can be done in real time (i.e. the amount of time it takes to write, or input, a character).

Referring to Fig. 5, a block diagram of the fast matching module (600) is shown. Generally, the input to the fast matching module is the above discussed preprocessed handwritten input that is described by a sequence of straight line strokes (125). The output of the fast matching module is a short list of candidate characters (625) that most probably corresponds and represents, or matches, the handwritten input (20).

Referring now to Fig. 6, a flow chart detailing the operation of the fast matching module is shown. In Fig. 6, the index i refers to the i th character of the preferred character set, and the index j refers to the j th template of the character.

The symbol Ti is the number of templates for the character whose index is i, and the quantity ms is the minimum string matching distance between the input and all the templates of one character. The minimum [illegible] a large number at the start of each [illegible] matching starts by attempting to match the input with the first template of the first character in the vocabulary (i.e. the character whose index is 1). The absolute difference d between the number of straight line strokes in the input and the number of straight line strokes in the jth template of character i is then computed (606). The difference d is compared against a threshold (608). The threshold can be fixed one or can be variable that depends on the number of strokes in the input and the template. A preferred threshold to use is computed as thresh = (number of strokes in the input + number of strokes in the template)/10 + 1. If the difference d is less than the threshold, a fast string matching distance s is computed between the input and the jth template of character i (610). The details of the obtaining the fast matching distance will be given in the following few paragraphs. The minimum string matching distance ms is updated (612) based on the newly computed fast string matching distance s (610). The steps 606, 608, 610, and 612 are repeated until all the Ti templates for character i are exhausted. Note that if the difference d is greater than the threshold, steps 610 and 612 are omitted and the next template for the character is considered. Once the minimum string matching distance between the input and the templates for character i has been computed, the current shortlist of characters that best match the input is updated (618). In addition to the shortlist of candidate characters that best match the input, the minimum fast string matching distances for matching the input with the templates for each character in the shortlist are also stored. The shortlist is sorted using the minimum fast string matching distances. Given a new character index and a corresponding minimum fast string

matching score, the current shortlist is updated by inserting the new character index into the shortlist at the location dictated by the new minimum fast string matching distance. If the new minimum fast string matching distance is greater than the last entry in the current list of minimum fast string matching distances, then the current shortlist does not need to be updated. After updating the current short list of candidate characters that best match the input, the next character in the vocabulary, if one such exists, is considered by starting again at step 604. After all the characters in the vocabulary have been considered, the short list of candidate characters that best match the input is forwarded to the detailed matching stage of the character matching module. In Fig. 6, the symbol M is used to denote the vocabulary size (i.e. M = number of characters in the vocabulary).

In a preferred method and embodiment of the present invention, the fast string matching in 610 is based on a single stroke feature. However, more than one stroke feature may be used for fast string matching. A preferred stroke feature to use for fast string matching is the angle of the stroke. Hence, the fast string matching distance is the distance between two one dimensional strings. A one dimensional string is simply a string of one dimensional quantities (like a single stroke feature). Note that the lengths of the two strings need not be the same. The technique used to compute the fast string matching distance is illustrated using a simple example. Figs. 7 and 8 show how the fast string matching distance between the strings S1 = [10, 25, 15, 90, 120] and S2 = [15, 5, 100, 140] is computed. The length of string S1 is 5 and the length of string S2 is 4. To begin with, the first element of string S1 is paired with the first element of string S2 and the current score is set to difference between the first element of the two strings. In this case, the current score is15 - 10 = 5 (see 802 in Fig. 8). At any given time, let the mth element of string S2 be paired with the nth element of string S1. To find the next best matching

pair of elements from the two strings, the three immediate neighboring pairs (m+1,n), (m+1,n+1), and (m,n+1) are compared (see 706). The pair that has the least distance is picked as the next matching pair and the the distance between the two elements in the new pair is added to the current fast string matching distance. In Fig. 8, the table 802 shows the first elements of the two strings forming a matching pair with a score of 5. To find the next matching pair of elements in the two strings, the three pairs (5,10), (5,25), and (15,25) are considered. Of the three pairs, the pair (5,10) has the least distance of 5. Hence the current pair is moved up and the fast string matching score is updated to 10 (see table 804 in Fig. 8). The processes of finding the best matching pairs in the two strings is repeated until the last element of string S1 is paired with the last element of string S2. These steps are illustrated in tables 806, 808, 810, and 812. The accumulated fast string matching distance when the last elements of the two strings are paired is the final fast string matching distance. In 812, the final string matching distance between the two strings S1 and S2 is 70. Table 814 in Fig. 8 shows the sequence of best matching pairs that was used to compute the fast string matching distance between the two strings S1 and S2.

Fig. 9 shows the computational blocks of the detailed matching module. The inputs to the detailed matching module are the preprocessed handwriten input (sequence of strokes) and the shortlist of candidate characters that is produced by the fast matching module. The output of the fast matching module is the final sorted list of candidate characters that best match the input. The detailed matching module comprises of two major computational blocks. The first block (902) finds a detailed matching distance between the input and the templates for all the characters included in the shortlist of candidate characters produced by the fast matching module. The output of 902 is a first detailed match list of candidate characters. The second detailed matching block (904) re-sorts

the first detailed match list of candidate characters to produces the final sorted list of candidate characters that best match the handwritten input.

Fig. 10 is a flow chart describing the dynamic programming based matching module (902). If Fig. 10, the index i refers to the ith entry in the shortlist of candidate characters produced by the fast matching module. The index of the character stored as the i th entry in the fast match short list is denoted by fi. The index j refers to the j th template of character fi and T(fi) is the number of templates stored for character fi. The symbol F is used to denote the size of the fast match short list. The quantity ms is the minimum dynamic programming matching distance between the input and all the templates of one character. The minimum dynamic programming matching distance is initialized to a large number at the start of each new character in the fast match shortlist (1004). Detailed matching starts by attempting to match the input with the first template of the first character in the fast match short list. A matching distance s is computed between the input and the jth template of character fi (see 1006) using the technique of dynamic programming. The technique of dynamic programming is known to one of ordinary skill in the art and can be found in the paper by Sakoe and Chiba. (H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", in Readings in Speech Recognition, A. Waibel and K-F Lee, editors. Morgan Kaufmann, San Mateo, California,USA. 1990.). In the present invention, dynamic programming is used to find a matching distance between two sequences of "strokes". The two sequences of "strokes" represent the prepreprocessed handwritten input and a stored template of a character. In a preferred method and embodiment of the present invention, a stroke is defined to be a straight line parametrized in some fashion. A preferred parametrization of the straight line strokes is by the four dimensional vector [mx, my, len, ang]

where mx is the x coordinate of the mid point of the stroke, my is the y coordinate of the mid point of the stroke, len is the length of the stroke, and ang is the angle made by the straight line stroke with respect to some reference axis. However, other definitions and parametrizations of strokes may be used.

In order to use the dynamic programming technique, the distance between two straight line strokes needs to be defined. A preferred stroke distance to use between two strokes parameterized as [mx1, my1, len1, ang1] and [mx2, my2, len2, ang2] is:

$$\text{stroke distance} = w_x \,\mathrm{abs}(mx1 - mx2) + w_2 \,\mathrm{abs}(my1 - my2) + w_l \,\mathrm{abs}(len1 - len2) + w_a \,\mathrm{cabs}(ang1 - ang2).$$

The quantities w_x, w_y, w_l and w_a are the weights associated with different dimensions of vector describing straight line stroke. The function abs(x) is the absolute value of x, and cabs(x) is the absolute value x assuming circular symmetry of x. Note that there is circular symmetry in the stroke angles, since 0 degrees is same as 360 degrees. In the preferred implementation, the quantities mx, my, len, and ang (that describe a straight line stroke) are all quantized to be between 0 and 255, so that a single byte (8 bits) can be used to store them. With the 8-bit quantization of the parameters describing a stroke, the preferred weights to use for computing the stroke distance is w_x = 1, w_y = 1, w_l = 1, and w_a = 4.

The minimum dynamic programming matching distance, ms, is updated (1008) based on the newly computed dynamic programming matching distance s (1006). The steps 1006 and 1008 are repeated until all the T(fi) templates for character fi are exhausted. Once the minimum dynamic programming matching distance between the input and the templates for character fi has been computed, the current first detailed match list of characters that best match the input is updated (1014). In addition to the first detailed match list of candidate

characters that best match the input, the minimum dynamic programming matching distances for matching the input with the templates for each character in the list are also stored. The first detailed match list is sorted using the minimum dynamic programming matching distances. Given a new character index and a corresponding minimum dynamic programming matching distance, the current first detailed match list is updated by inserting the new character index into the first detailed match list at the location dictated by the new minimum dynamic programming matching distance. If the new minimum dynamic programming matching distance is greater than the last entry in the current list of minimum dynamic programming matching distances, then the current first detailed match list does not need to be updated. After updating the current first detailed match list of candidate characters that best match the input, the next character in the fast match list, if one such exists, is considered by starting again at step 1004. After all the characters in the fast match list have been considered, the first detailed match list of candidate characters that best match the input is forwarded to the module that resorts the first detailed match list of candidate characters in order to produce the final sorted list of candidate characters that best match the input.

Fig. 11 is a flow chart describing the module that re-sorts the first detailed match list of candidate characters. If Fig. 11; [illegible] refers to the i th entry in the first detailed match [illegible] produced by the dynamic [illegible] matching module. The index of the character stored as the i th entry in the first detailed match list is denoted by li. The index j refers to the j th template of character li and T(li) is the number of templates stored for character li. The symbol L is used to denote the size of the first detailed match list. The quantity ms is the minimum weighted dynamic programming matching distance between the input and all the templates of one character. The minimum weighted dynamic programming matching distance is initialized to a

large number at the start of each new character in the first detailed match list (1104). Re-sorting the first detailed match list starts by attempting to compute a minimum weighted dynamic programming matching distance between the input and the first template of the first character in the first detailed match list. A weighted dynamic programming matching distance s is computed between the input and the jth template of character li (see 1106) using the technique of dynamic programming and then weighting individual stroke errors in order to produce the final matching distance. Traditional dynamic programming m[illegible] which gives the pairing of the input [illegible] the template. The concept of best path is known to one of ordinary skill in the art. The normal dynamic programming matching distance is simply the sum of the inter-stroke distances along the best path. In order to get the weighted dynamic programming matching distance, a weighted sum of the inter-stroke distance along the best path is used. The weight used for each inter-stroke distance in the best path is weight stored for each template stroke. The rationale for using a weighted dynamic programming matching distance is that some strokes in a handwritten character may be more consistent than other strokes when multiple instance of the same character are considered Hence the more consistent strokes need to be weighted more in order to get robust recognition of handwritten input. The minimum weighted dynamic programming matching distance, ms, is updated (1108) based on the newly computed weighted dynamic programming matching distance s (1106). The steps 1106 and 1108 are repeated until all the T(li) templates for character li are exhausted. Once the minimum weighted dynamic programming matching distance between the input and the templates for character li has been computed, the current sorted list of characters that best match the input is updated (1114). In addition to the sorted match list of candidate

characters that best match the input, the minimum weighted dynamic programming matching distances for matching the input with the templates for each character in the list are also stored. The final sorted match list is sorted using the minimum weighted dynamic programming matching distances. Given a new character index and a corresponding minimum weighted dynamic programming matching distance, the current sorted match list is updated by inserting the new character index into the sorted match list at the location dictated by the new minimum weighted dynamic programming matching distance. If the new minimum weighted dynamic programming matching distance is greater than the last entry in the current list of minimum weighted dynamic programming matching distances, then the current sorted match list does not need to be updated. After updating the current sorted match list of candidate characters that best match the input, the next character in the first detailed match list, if one such exists, is considered by starting again at step 1104. After all the characters in the first detailed match list have been considered, the sorted match list of candidate characters becomes the final sorted list of candidate characters that best match the input

Those skilled in the art will find many embodiments of the present invention to be useful. One obvious advantage is ease of data, or text, input over traditional key-board entry methods, including the obvious advantage of the ease of entry [illegible] handwritten input into printed data, or [illegible] recognition of handwritten [illegible] character, data point, or other unitary identifying graphically representation, that is a [illegible] data points, or other graphical representation.

It will be apparent to those skilled in the art that the disclosed invention may be modified in numerous ways and may assume many embodiments other than the preferred forms particularly set out and described above. Accordingly, it

is intended by the appended claims to cover all modifications of the present invention that fall within the true spirit and scope of the present invention and its equivalents.

What is claimed

1. A method, comprising the steps of:

receiving handwritten input as data representing a sequence strokes;

determining a plurality of candidate symbols from stored templates that are likely matches for the handwritten input by comparing one or more stroke parameters between the sequence of strokes representing the handwritten input and the sequence of strokes for a plurality of symbols from the templates; and

determining one or more recognized symbols that are likely matches for the handwritten input by comparing two or more stroke parameters between the sequence of strokes representing the handwritten input and the sequence of strokes for each symbol in the plurality of candidate symbols that are likely matches for the handwritten input.

2. A method, comprising the steps of:

processing handwritten input as a sequence of handwritten strokes to provide data representing a sequence of straight strokes;

determining a plurality of candidate symbols from stored templates that are likely matches for the handwritten input by comparing one or more stroke parameters between the sequence of straight strokes representing the handwritten input and the sequence of strokes for a plurality of symbols from the templates; and

determining one or more recognized symbols that are likely matches for the handwritten input by comparing two or more stroke parameters between the sequence of straight strokes representing the handwritten input and the sequence of strokes for each symbol in the plurality of candidate symbols that are likely matches for the handwritten input.

3. A method, comprising the steps of:

receiving handwritten input as data representing a sequence strokes;

determining an angle parameter for each of the sequence of strokes, the angle parameter representing an angle of the stroke from a reference axis;

determining a plurality of candidate symbols from stored templates that are likely matches for the handwritten input by comparing the angle parameter between the sequence of strokes representing the handwritten input and the sequence of strokes for a plurality of symbols from the templates; and

determining one or more recognized symbols that are likely matches for the handwritten input by comparing other stroke parameters between the sequence of strokes representing the handwritten input and the sequence of strokes for each symbol in the plurality of candidate symbols that are likely matches for the handwritten input.

4. A method, comprising the steps of:

receiving handwritten input as data representing a sequence strokes;

determining stroke parameters for each of the sequence of strokes, the stroke parameters selected from the group of parameters consisting of:

an angle parameter representing an angle of the stroke from a reference axis;

a x coordinate midpoint parameter representing the x coordinate of the midpoint of the stroke;

a y coordinate midpoint parameter representing the y coordinate of the midpoint of the stroke;

a length parameter representing stroke length;

determining a plurality of candidate symbols from stored templates that are likely matches for the handwritten

input by comparing one stroke parameter between the sequence of strokes representing the handwritten input and the sequence of strokes for a plurality of symbols from the templates; and

determining one or more recognized symbols that are likely matches for the handwritten input by comparing two or more stroke parameters between the sequence of strokes representing the handwritten input and the sequence of strokes for each symbol in the plurality of candidate symbols that are likely matches for the handwritten input.

5. An apparatus, comprising

a digitizing tablet for receiving handwritten input as a sequence of strokes;

a memory having data and instructions stored therein and having a plurality of templates representing characters or symbols some of which may correspond to the handwritten input;

a processor for processing the data or instructions memory to provide a plurality of candidate symbols by comparing at least one stroke parameter between the sequence of strokes representing the handwritten input and a sequence of strokes for one or more characters or symbols in the memory, and for providing a selectable plurality of recognized symbols by comparing two or more stroke parameters between the sequence of strokes representing the handwritten input and the sequence of strokes for each of the plurality of candidate symbols.

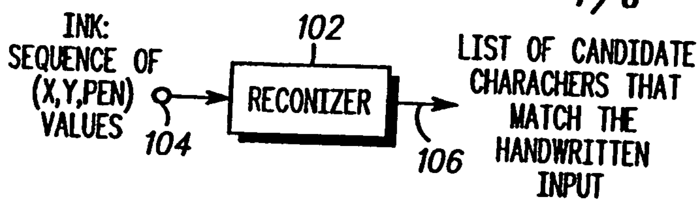


FIG.1

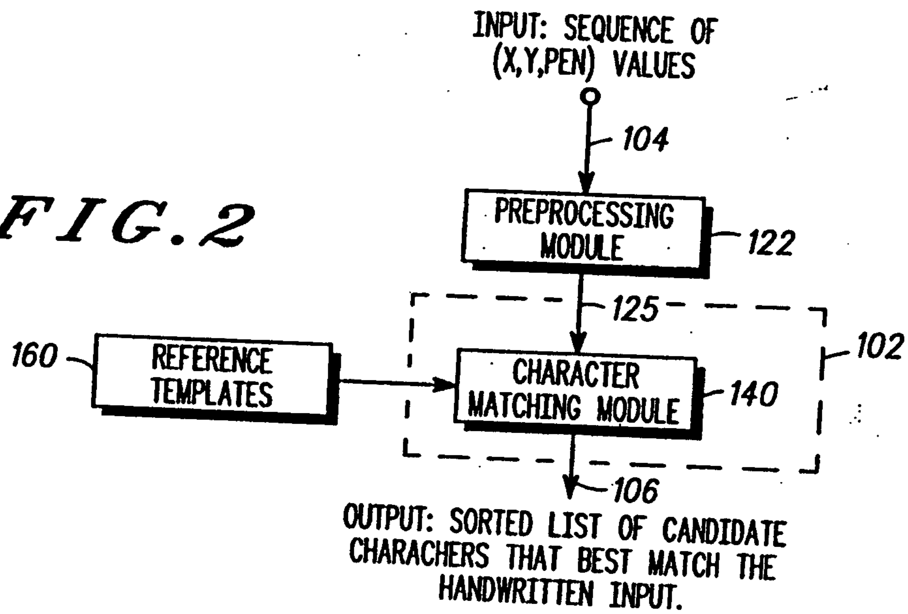
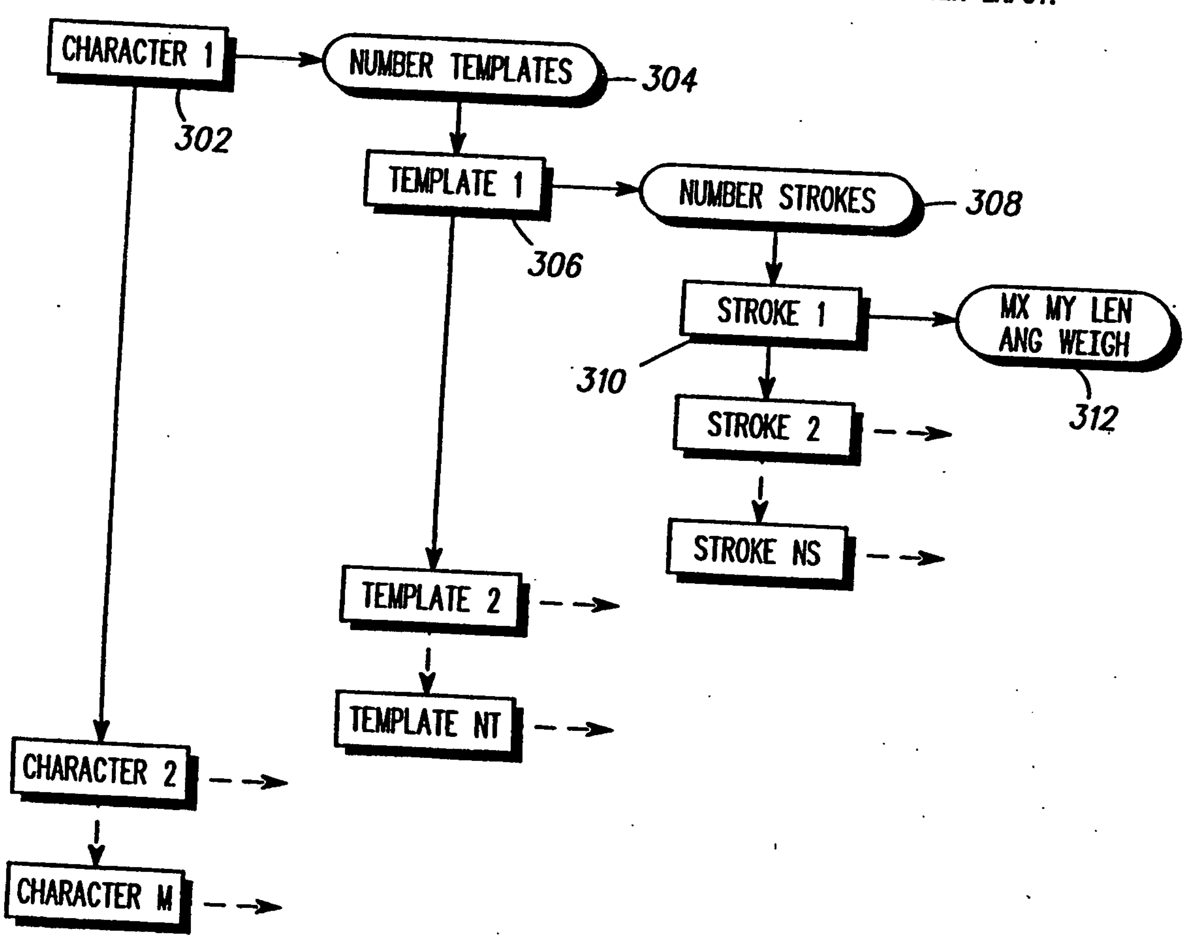


FIG.2

FIG.3



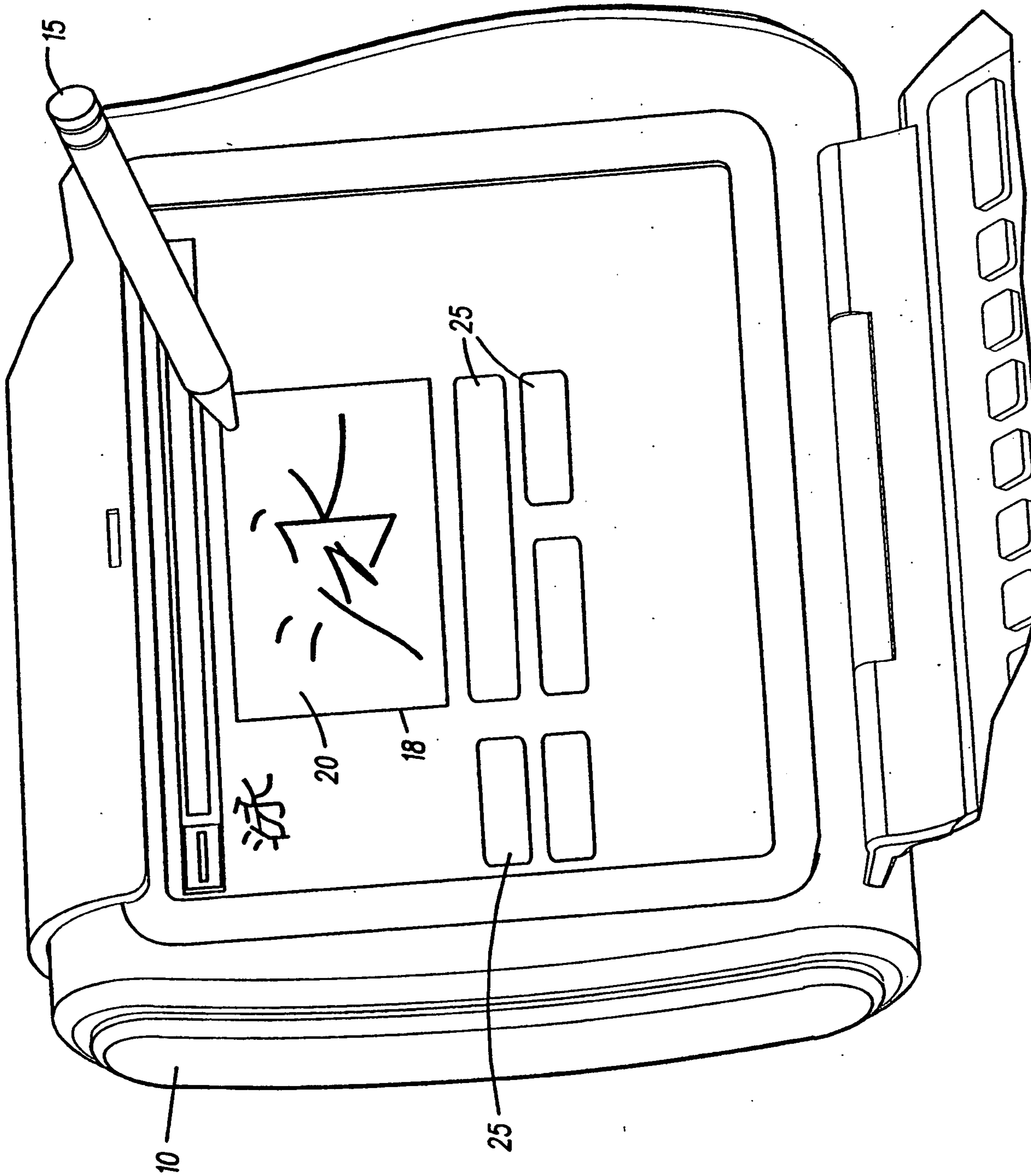


FIG. 1A

FIG. 4

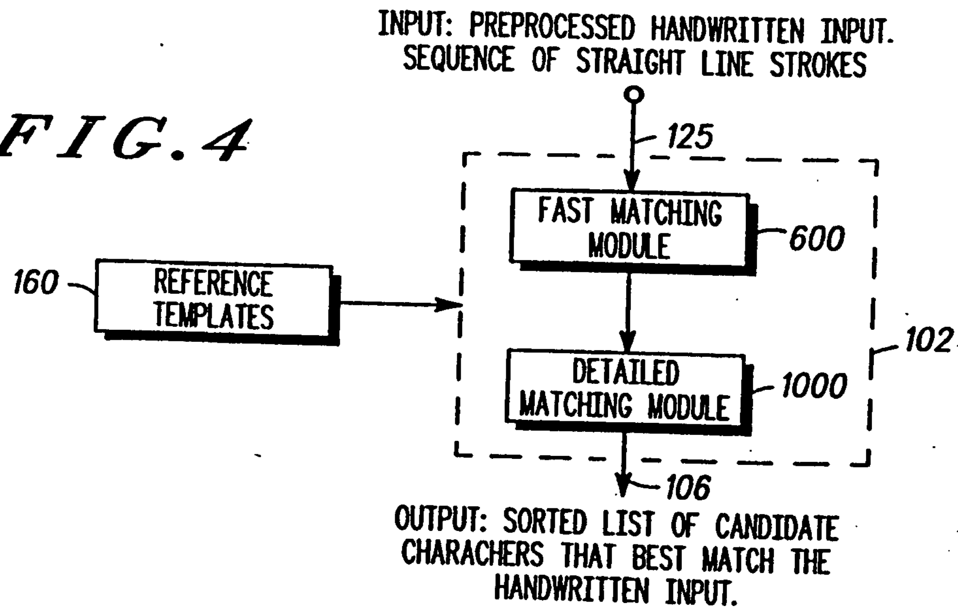


FIG. 5

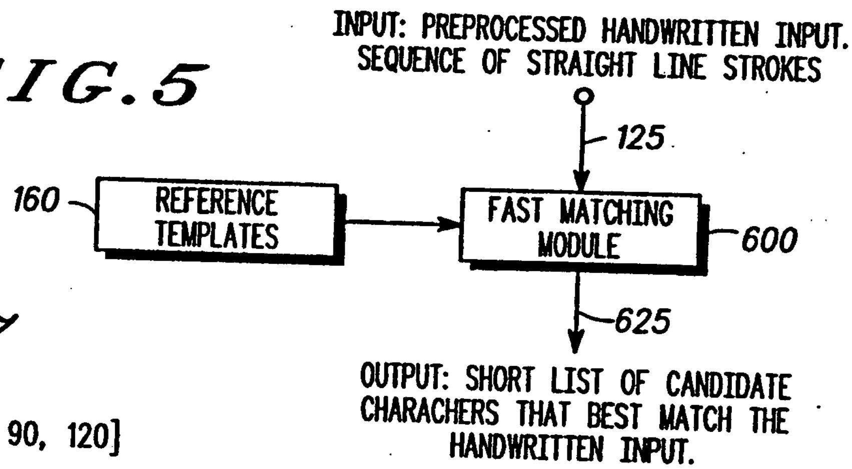


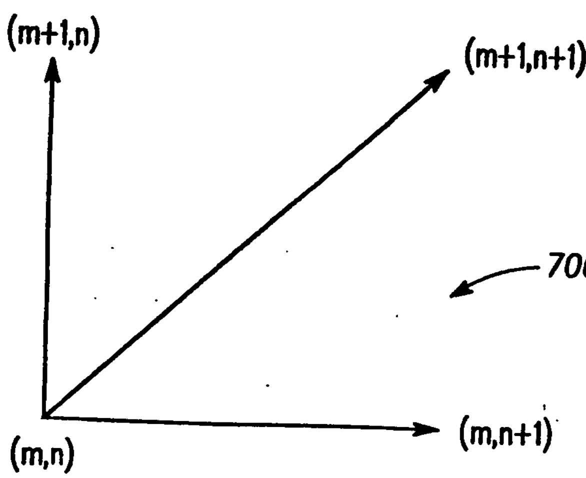
FIG. 7

702

S1 = [10, 25, 15, 90, 120]

S2 = [15, 5, 100, 140]

704



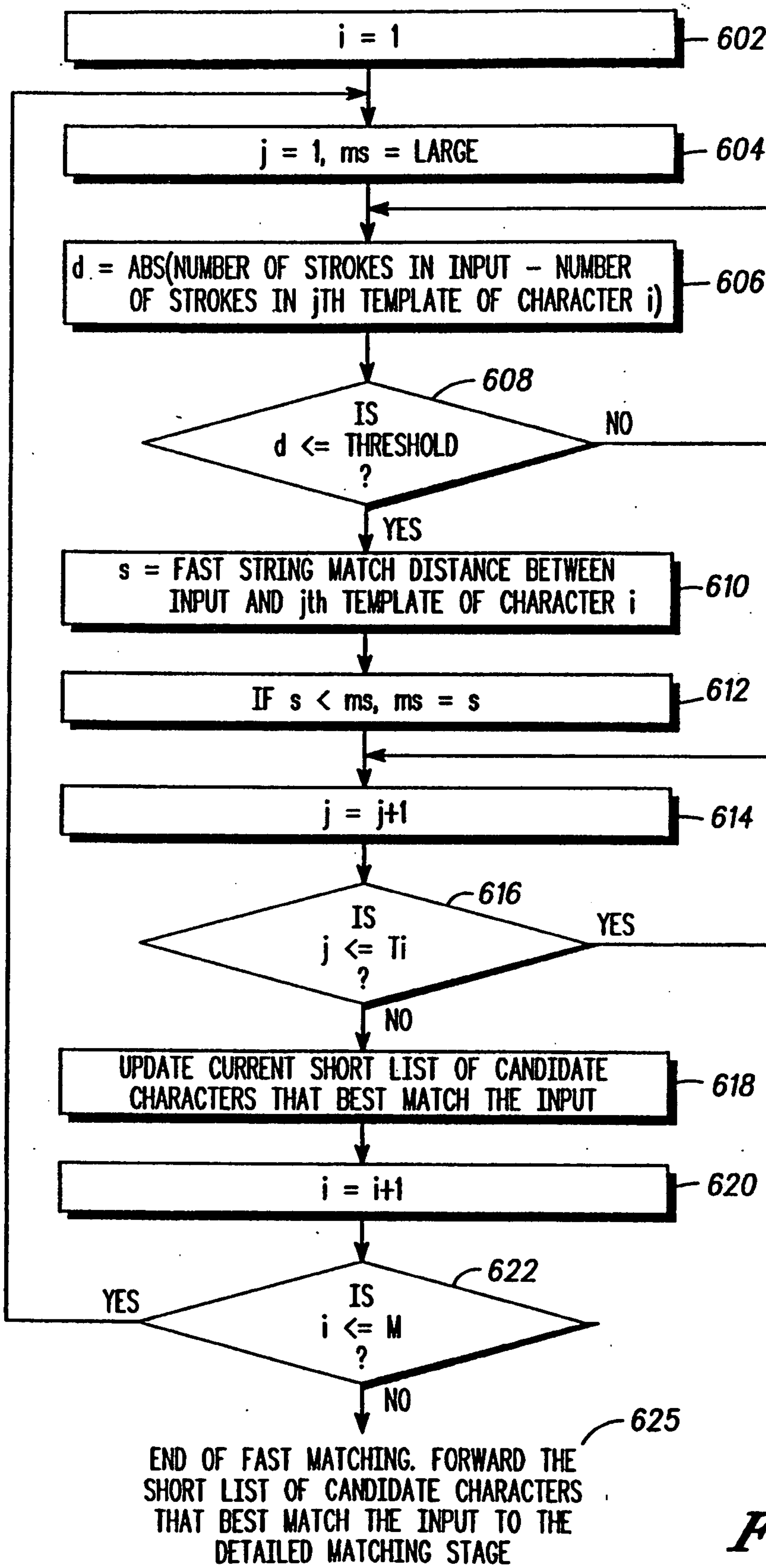

i = 1
602
j = 1, ms = LARGE
604
d = ABS(NUMBER OF STROKES IN INPUT - NUMBER OF STROKES IN jTH TEMPLATE OF CHARACTER i)
606
608
IS d <= THRESHOLD ?
NO
YES
s = FAST STRING MATCH DISTANCE BETWEEN INPUT AND jth TEMPLATE OF CHARACTER i
610
IF s < ms, ms = s
612
j = j+1
614
616
IS j <= Ti ?
YES
NO
UPDATE CURRENT SHORT LIST OF CANDIDATE CHARACTERS THAT BEST MATCH THE INPUT
618
i = i+1
620
622
IS i <= M ?
YES
NO
625
END OF FAST MATCHING. FORWARD THE SHORT LIST OF CANDIDATE CHARACTERS THAT BEST MATCH THE INPUT TO THE DETAILED MATCHING STAGE

FIG. 6

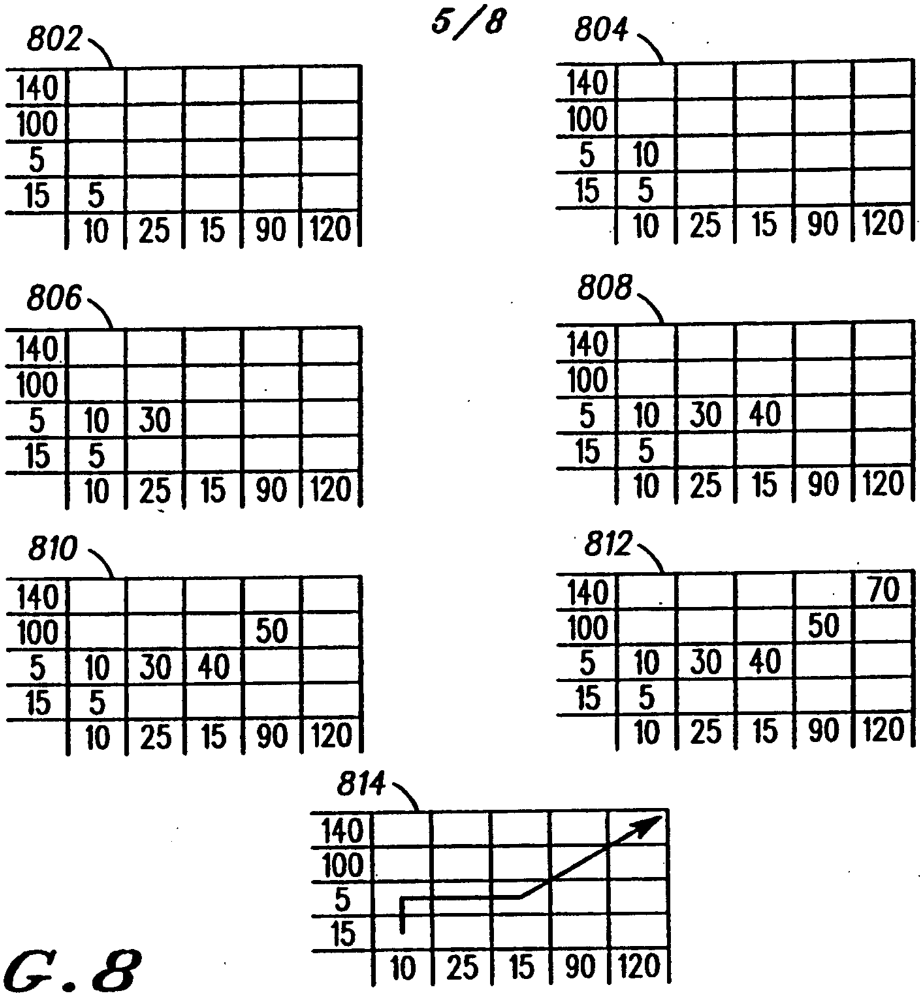

802
804
806
808
810
812
814
140
100
5
15
10
25
15
90
120
30
40
50
70

FIG. 8

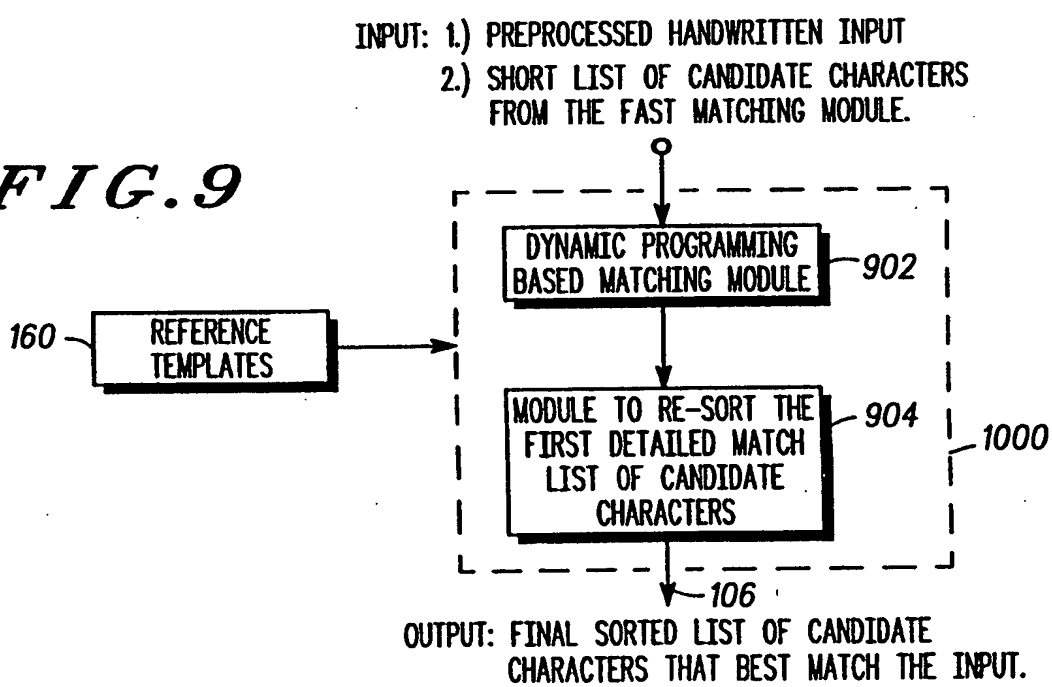

INPUT: 1.) PREPROCESSED HANDWRITTEN INPUT
2.) SHORT LIST OF CANDIDATE CHARACTERS FROM THE FAST MATCHING MODULE.
DYNAMIC PROGRAMMING BASED MATCHING MODULE
902
160
REFERENCE TEMPLATES
MODULE TO RE-SORT THE FIRST DETAILED MATCH LIST OF CANDIDATE CHARACTERS
904
1000
106
OUTPUT: FINAL SORTED LIST OF CANDIDATE CHARACTERS THAT BEST MATCH THE INPUT.

FIG. 9

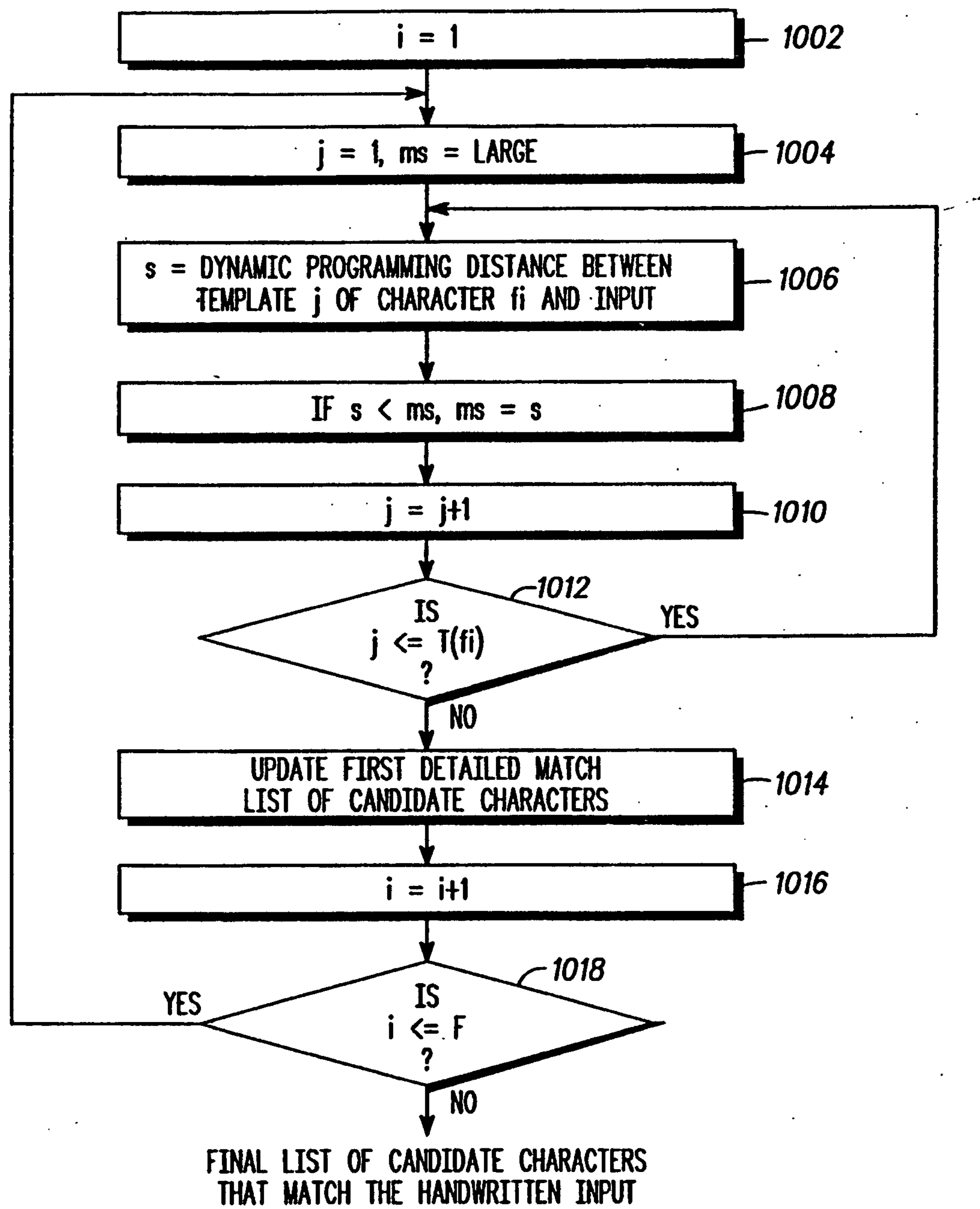
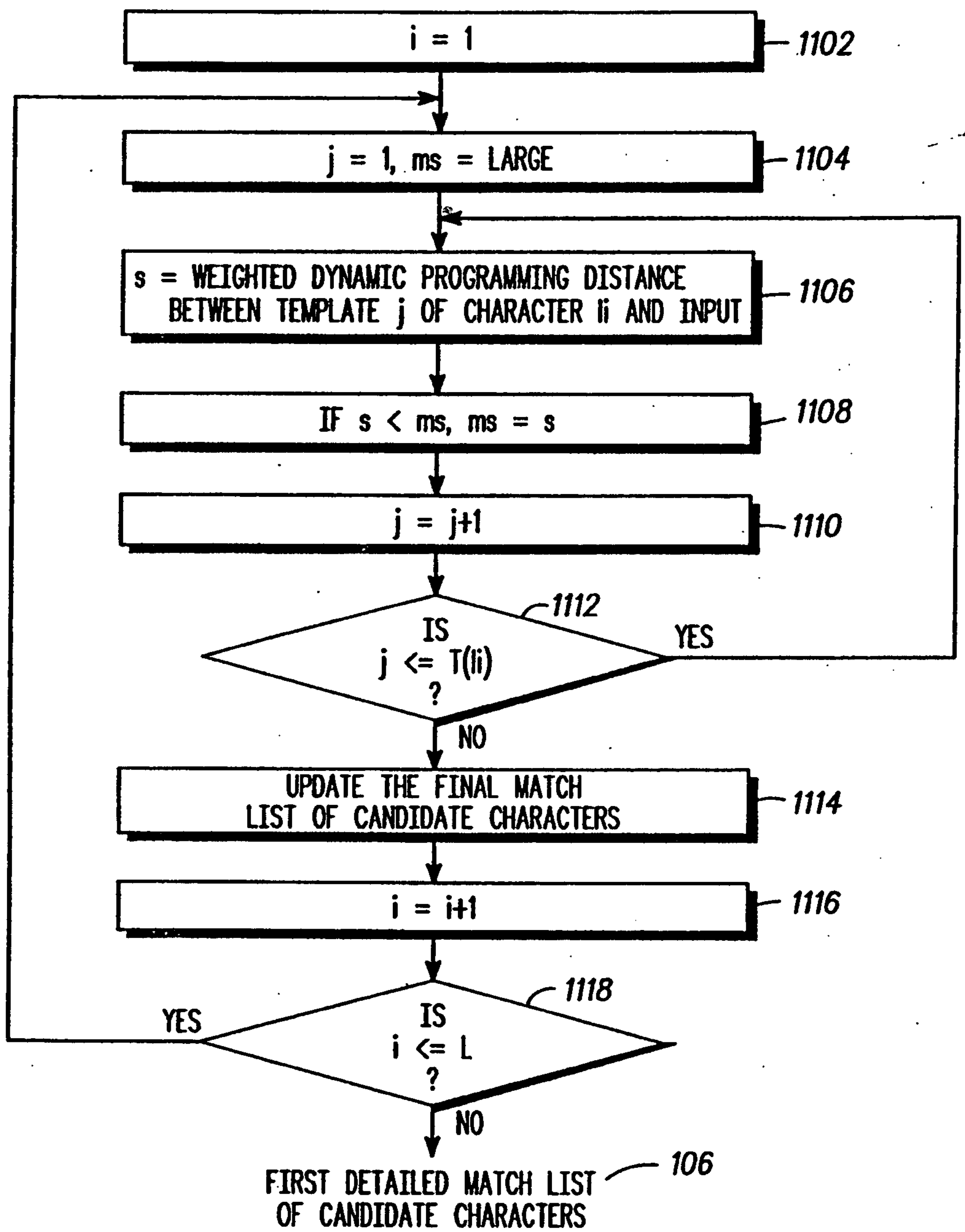

i = 1
1002
j = 1, ms = LARGE
1004
s = DYNAMIC PROGRAMMING DISTANCE BETWEEN TEMPLATE j OF CHARACTER fi AND INPUT
1006
IF s < ms, ms = s
1008
j = j+1
1010
IS j <= T(fi) ?
1012
YES
NO
UPDATE FIRST DETAILED MATCH LIST OF CANDIDATE CHARACTERS
1014
i = i+1
1016
IS i <= F ?
1018
YES
NO
FINAL LIST OF CANDIDATE CHARACTERS THAT MATCH THE HANDWRITTEN INPUT

FIG.10



i = 1
1102
j = 1, ms = LARGE
1104
s = WEIGHTED DYNAMIC PROGRAMMING DISTANCE BETWEEN TEMPLATE j OF CHARACTER li AND INPUT
1106
IF s < ms, ms = s
1108
j = j+1
1110
1112
IS j <= T(li) ?
YES
NO
UPDATE THE FINAL MATCH LIST OF CANDIDATE CHARACTERS
1114
i = i+1
1116
1118
IS i <= L ?
YES
NO
FIRST DETAILED MATCH LIST OF CANDIDATE CHARACTERS
106

FIG.11

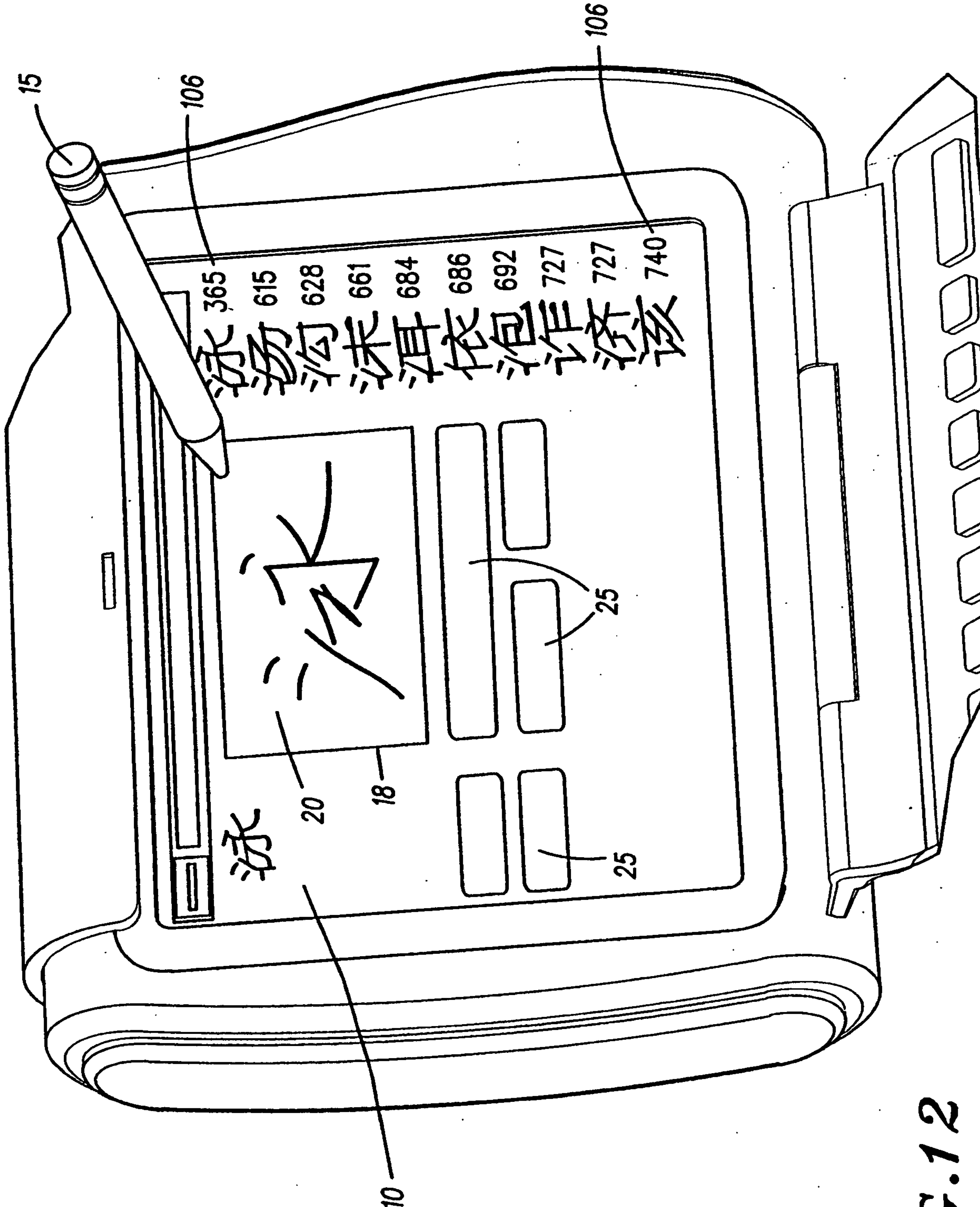

15
106
106
106
365
615
628
661
684
686
692
727
727
740
25
25
20
18
10

FIG.12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US96/05884

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06K 9/18
US CL :382/185, 227
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 382/185-189, 224, 227; 178/18-20

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

none

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

none

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X ------ Y	US, A, 4,516,262 (SAKURAI) 07 May 1985, Abstract, Figs. 1, 2, 7-9, col. 1, line 31 to col. 3, line 54.	1, 2, 5 --------- 3, 4
Y	US, A, 4,685,142 (OOI ET AL) 04 August 1987, Abstract, Figs. 2, 4, 6a-9, col. 2, lines 1-26, and col. 4, line 27 to col. 5, line 48.	3, 4
Y	US, A, 4,607,386 (MORITA ET AL) 19 August 1986, Abstract, Figs. 1-4, col. 1, lines 29-58, and col. 2, line 20 to col. 4, line 25.	3, 4

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be part of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

05 AUGUST 1996

Date of mailing of the international search report

17 SEP 1996

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
JOSEPH MANCUSO
Telephone No. (703) 305-8503